ETSI TS 103 146-3 V1.1.1 (2016-02)

Reconfigurable Radio Systems (RRS);
Mobile device information models and protocols;
Part 3: Unified Radio Application Interface (URAI)

Reference
DTS/RRS-0247

Keywords
architecture, mobile, SDR, software

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Reconfigurable Radio Systems (RRS).

The present document is part 3 of a multi-part deliverable covering the Mobile Device Information Models and Protocols, as identified below:

Part 1: "Multiradio Interface (MURI)";

Part 2: "Reconfigurable Radio Frequency Interface (RRFI)";

Part 3: "Unified Radio Application Interface (URAI)";

Part 4: "Radio Programming Interface (RPI)".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The scope of the present document is to define an information model and protocol for unified radio application interface for mobile device reconfiguration. The work is based on the Use Cases defined in ETSI TR 102 944 [i.1], on the system requirements defined in ETSI EN 302 969 [1] and on the radio reconfiguration related architecture for mobile devices defined in ETSI EN 303 095 [i.2] and on the mobile device information models and protocols related Multiradio Interface defined ETSI TS 103 146-1 [i.3].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI EN 302 969 (V1.2.1): "Reconfigurable Radio Systems (RRS); Radio Reconfiguration related Requirements for Mobile Devices".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 102 944: "Reconfigurable Radio Systems (RRS); Use Cases for Baseband Interfaces for Unified Radio Applications of Mobile Device".

[i.2] ETSI EN 303 095 (V1.2.1): "Reconfigurable Radio Systems (RRS); Radio Reconfiguration related Architecture for Mobile Devices".

[i.3] ETSI TS 103 146-1: "Reconfigurable Radio Systems (RRS); Mobile Device Information Models and Protocols; Part 1: Multiradio Interface (MURI)".

[i.4] ETSI TS 103 146-2: "Reconfigurable Radio Systems (RRS); Mobile Device Information Models and Protocols; Part 2: Reconfigurable Radio Frequency Interface (RRFI)".

[i.5] ETSI TR 102 839: "Reconfigurable Radio Systems (RRS); Multiradio Interface for Software Defined Radio (SDR) Mobile Device Architecture and Services".

[i.6] IEEE 1900.4-2009™: "IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks".

[i.7] Recommendation ITU-T X.680: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

association: logical communication link to a Radio Access Network or a peer equipment

NOTE 1: Typically, some control signalling is necessary to maintain the association. No user data transfer may occur with only an association present, but a data flow may be established into an association for this purpose.

NOTE 2: Peer equipment is any communication counterpart of a reconfigurable mobile device. It can be reached by establishing a logical communication link (i.e. an association) between the reconfigurable mobile device and peer equipment.

channel: designated part of the information transfer capability having specified characteristics, provided at the user network interface

NOTE: It is the over-the-air wireless propagation channel which is used to convey an information signal from transmitter to receiver. This definition is specified in ETSI EN 303 095 [i.2].

communication services layer: layer related to communication services supporting generic applications

NOTE: A communication services layer supports generic applications like Internet access. In the present document, it consists of Administrator, Mobility Policy Manager (MPM), Networking stack and Monitor.

link: connection from one location to another through a given Radio Access Technology for the purpose of transmitting and receiving digital information

Radio Application (RA): software which enforces the generation of the transmit RF signals or the decoding of the receive RF signals

NOTE 1: The Software is executed on a particular radio platform or an RVM as part of the radio platform.

NOTE 2: RAs might have different forms of representation. They are represented as:

- source codes including Radio Library calls of Radio Library native implementation and Radio HAL calls;
- IRs including Radio Library calls of Radio Library native implementation and radio HAL calls;
- Executable codes for a particular radio platform.

radio computer: part of mobile device hardware working under ROS control and on which RAs are executed

NOTE: A Radio Computer typically includes programmable processors, hardware accelerators, peripherals, etc. RF part is considered to be part of peripherals.

Radio Control Framework (RCF): control framework which, as a part of the OS, extends OS capabilities in terms of radio resource management

NOTE: RCF is a control framework which consists of Configuration Manager (CM), Radio Connection Manager (RCM), Flow Controller (FC) and Multiradio Controller (MRC). The Resource Manager (RM) is typically part of OS.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
BLER	Block Error Rate
CM	Configuration Manager
CSL	Communication Services Layer

FC	Flow Controller
ID	Identification
MD	Mobile Device
MPM	Mobility Policy Manager
MRC	Multiradio Controller
MURI	Multiradio Interface
OS	Operating System
RA	Radio Application
RAN	Radio Access Network
RAP	Radio Application Package
RAT	Radio Access Technology
RCF	Radio Control Framework
RCM	Radio Connection Manager
RF	Radio Frequency
RM	Resource Manager
ROS	Radio Operating System
RPI	Radio Programming Interface
RRFI	Reconfigurable Radio Frequency Interface
SINR	Signal to Interference plus Noise Ratio
UML	Unified Modelling Language
URA	Unified Radio Applications
URAI	Unified Radio Application Interface

4 Introduction

A reconfigurable MD is capable of running multiple radios simultaneously and of changing the set of radios by loading new Radio Application Package (RAP). All Radio Applications (RAs) are called Unified Radio Applications (URAs) when they exhibit a common behaviour from the reconfigurable MD's point of view [i.2]. In order to run multiple URAs, the reconfigurable MD will include Communication Services Layer (CSL), Radio Control Framework (RCF), Radio Platform and 4 sets of interfaces for their interconnection.

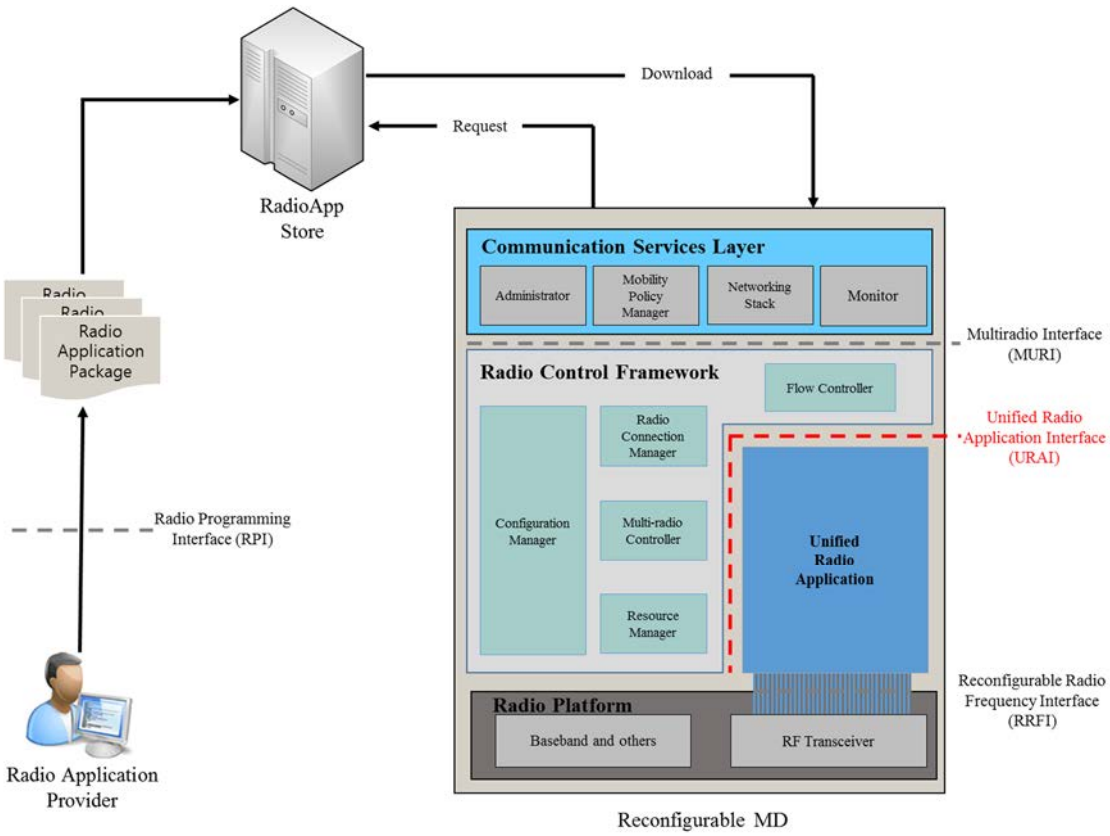

Figure 4.1: Four sets of interfaces for Reconfigurable MD

Figure 4.1 illustrates the Reconfigurable MD architecture with the 4 sets of interfaces, i.e.:

- MURI for interfacing CSL and RCF [i.3];
- RRFI for interfacing URA and RF Transceiver [i.4];
- URAI for interfacing URA and RCF which is the scope of the present document ;
- RPI for allowing an independent and uniform production of RAs [i.5].

The present document defines UARI.

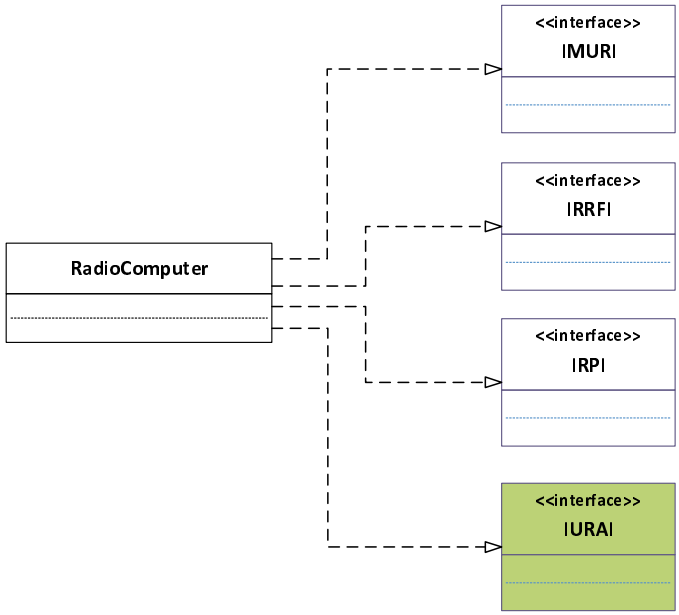

Figure 4.2: UML class diagram for Radio Computer interfaces

Figure 4.2 illustrates UML class diagram for Radio Computer interfaces. The reconfigurable MD may be seen as a Radio Computer where individual URAs are engineered as software entities [i.2].

The present document is organized as follows:

- Clause 5 describes the system identification;
- Clause 6 describes the notational tool for defining both information model classes and interface classes;
- Clause 7 describes the information model for radio computer; and
- Clause 8 describes the interface definition.

While UML is used for defining the information model and protocol related to URAI, other modelling languages could be used as well.

5 System Identification

5.1 Radio Computer Structure

Figure 5.1 illustrates how RCF and URA interact with each other using URAI.

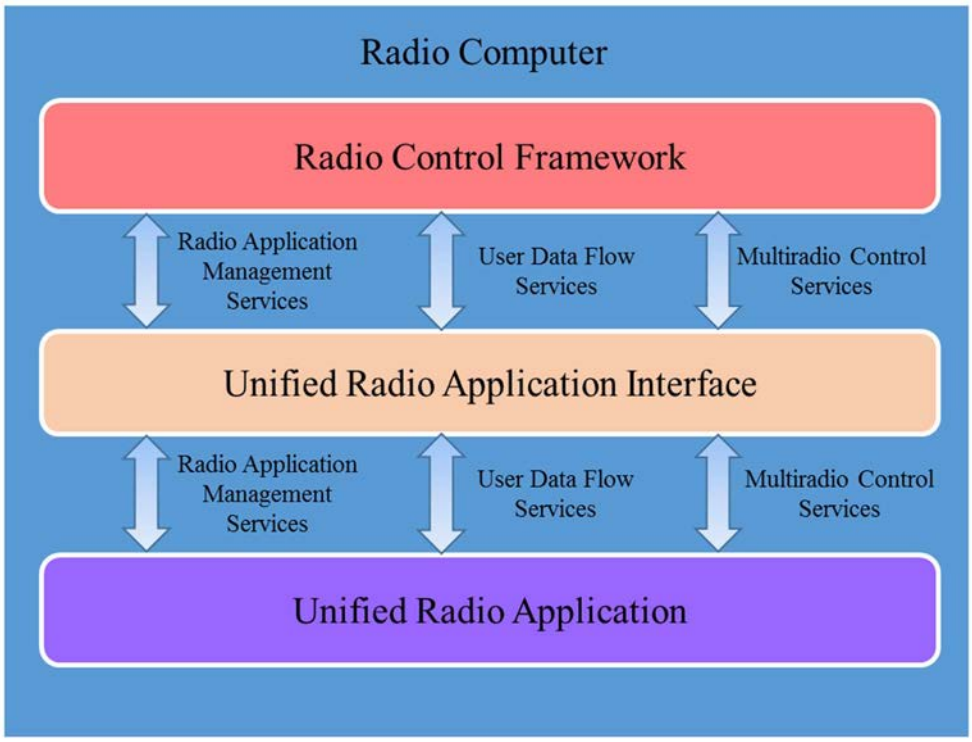

Figure 5.1: Interconnection between RCF and URA using URAI for Reconfigurable MD.

As shown in figure 5.1, URAI supports 3 kinds of services:

- **Radio Application Management Services**

These services are used by Radio Connection Manager (RCM) which is included in the RCF, to control URA functions such as reporting of discovered Peer Equipments, creating/terminating association with Peer Equipment, starting/stopping communication with Peer Equipment, etc.

- **User Data Flow Services**

These services are used by Flow Controller (FC) which is included in the RCF, to transmit user data to URA, or used by URA to transmit received user data to FC. These services also include management of data flow, which is provided by FC.

- **Multiradio Control Services**

These services are used by Multiradio Controller (MRC) which is included in RCF, to manage spectral resource usage.

The RCF and URA are defined in ETSI EN 303 095 [i.2].

5.2 URAI System Requirement Mapping

The Radio Computer components above described shall support the URAI system requirements shown in table 5.1 and described in clause 6 of ETSI EN 302 969 [1].

Table 5.1: Mapping of Radio Computer Components to the system requirements described in ETSI EN 302 969 [1]

Entity/Component/Unit	System Requirements [1]	Comments
Flow Controller	R-FUNC-RAT-05	If a reconfigurable MD allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), various independent data flows should be maintained simultaneously. The requirement is described in clause 6.1.5 of ETSI EN 302 969 [1].
	R-FUNC-RA-04	Radio Applications should support the function of transferring receive (Rx)/transmit (Tx) data to/from the networking stack. The requirement is described in clause 6.2.4 of ETSI EN 302 969 [1].
Multiradio Controller	R-FUNC-RAT-01	A reconfigurable MD should support parallel connections to more than one Radio Access Technology. The requirement is described in clause 6.1.1 of ETSI EN 302 969 [1].
	R-FUNC-RAT-02	If a reconfigurable MD allows parallel connections to RATs, (in alignment to R-FUNC-RAT-01), in-device coexistence functionalities shall be implemented. The requirement is described in clause 6.1.2 of ETSI EN 302 969 [1].
	R-FUNC-RAT-03	If a reconfigurable MD allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), seamless handover of data streams from one RAT to another RAT should be implemented. The requirement is described in clause 6.1.3 of ETSI EN 302 969 [1].
	R-FUNC-RAT-06	If a reconfigurable MD allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), Link Adaptation techniques across multiple RATs should be implemented. The requirement is described in clause 6.1.6 of ETSI EN 302 969 [1].
	R-FUNC-MDR-03	The radio configuration of a reconfigurable MD shall be realized with the activation of Radio Applications (RA) and, if necessary, changing parameters of the activated RAs. The requirement is described in clause 6.4.3 of ETSI EN 302 969 [1].
Radio Connection Manager	R-FUNC-RAT-04	If policies are applied to a reconfigurable MD, the link selection functionality in the reconfigurable MD shall meet the related conditions. The requirement is described in clause 6.1.4 of ETSI EN 302 969 [1].
	R-FUNC-RAT-05	If a reconfigurable MD allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), various independent data flows should be maintained simultaneously. The requirement is described in clause 6.1.5 of ETSI EN 302 969 [1].
	R-FUNC-RAT-06	If a reconfigurable MD allows parallel connections to RATs (in alignment to R-FUNC-RAT-01), Link Adaptation techniques across multiple RATs should be implemented. The requirement is described in clause 6.1.6 of ETSI EN 302 969 [1].
	R-FUNC-RA-03	Reconfigurable MDs should support concurrent execution of Radio Applications. The requirement is described in clause 6.2.3 of ETSI EN 302 969 [1].
	R-FUNC-MDR-03	The radio configuration of a reconfigurable MD shall be realized with the activation of Radio Applications (RA) and, if necessary, changing parameters of the activated RAs. The requirement is described in clause 6.4.3 of ETSI EN 302 969 [1].

6 Notational Tools

6.1 Notational Tool for Information Model Classes

Table 6.1 shows a template for defining information model classes [i.6]. Each information model class is defined in clause 7.2 in accordance with the template shown in table 6.1.

NOTE: ASN.1 is used throughout the present document for abstract type definitions; however, alternative ways are possible and are not excluded.

Table 6.1: Template for defining Information Model Classes

<table>
<tr><td colspan="4">Class<Class name>[(abstract class)]</td></tr>
<tr><td colspan="4"><Description of the class></td></tr>
<tr><td>DERIVED FROM</td><td colspan="3"><List of super-classes></td></tr>
<tr><td colspan="4">ATTRIBUTES</td></tr>
<tr><td><Attribute name>
[<optional>]</td><td>Value type:
<Attribute value type></td><td>Possible access:
<Attribute access qualifier></td><td>Default value:
<Default value></td></tr>
<tr><td colspan="4"><Description of the attribute></td></tr>
<tr><td>CONTAINED IN</td><td colspan="3"><List of classes, whose instances may contain an instance of this class. If this class is an abstract class, that is, it is used for further refinement only and will never be instantiated, then this list is empty.></td></tr>
<tr><td>CONTAINS</td><td colspan="3"><List of classes, whose instances may be contained in an instance of this class. Constraints used are:
• [*] - zero or more instances,
• [+] - one or more instances,
• [<n>] - exactly n instances,
• [<m> - <n>] - not less than m and not more than n instances.></td></tr>
<tr><td>SUPPORTED EVENTS</td><td colspan="3"><List of event names that are detected by this class and lead potentially to a corresponding event report.></td></tr>
</table>

Further details on the template in table 6.1 are given below.

- <Class name> is the name of the Class as it appears in the corresponding model. Additional information is also included in case the class in question has been specified as an abstract one.
- DERIVED FROM field identifies the super class of the class in case of sub-classing.
- ATTRIBUTES field describes the attributes that have been defined in the class. More specifically:
  - <Attribute name> identifies the name of an attribute, as it is included in the class definition.
  - <Attribute value type> holds the type of the attribute specified in Abstract Syntax Notation One (ASN.1). Details related to the ASN.1 module are specified in annex A.
  - <Attribute access qualifier> provides information about the level of accessibility of the attribute. This may include: 'Read', 'Write', 'Read-Write', 'Add-Remove' (for list-type attributes), 'Read-Add-Remove', and 'None' (for internal access only).
- CONTAINED IN field includes a list of classes whose instances may contain an instance of this class; containment is a strong aggregation relationship, that is, a contained instance is for its lifetime bound to its container object and it is contained only in this one container.
- CONTAINS field provides a list of classes whose instances may be contained in an instance of the class in question.
- SUPPORTED EVENTS field includes a list of event names that are detected by this class and lead potentially to a corresponding event report.

6.2 Notational Tool for Interface Classes

Table 6.2 shows a template for defining interface classes for URAI. Each interface class for URAI will be defined in clause 8.5 in accordance with the template shown in table 6.2.

Table 6.2: Template for defining Interface Classes

Class<Class name>*[(abstract class)]*		
<Description of the class>		
OPERATIONS		
<Operation name>	*Return type:* <Operation return type>	*Value type:* <Operation value type>
<Description of the operation>		

The template fields in table 6.2 are described below.

- <Class name> is the name of the Class as it appears in the corresponding model. Additional information is also included in case the class in question has been specified as an abstract one.
- OPERATIONS field describes the operations that have been defined in the class. More specifically:
  - <Operation name> identifies the name of an operation, as it is included in the class definition.
  - <Return type> identifies the type of return value at the corresponding operation. Details related to the ASN.1 module are specified in annex B.
  - <Value type> identifies the access levels for member functions: public, private, protected.

7 Information Model for Radio Computer

7.1 Radio Computer

Figure 7.1 shows the UML class diagram for Radio Computer classes related to URAI which are required to support Software Reconfiguration.

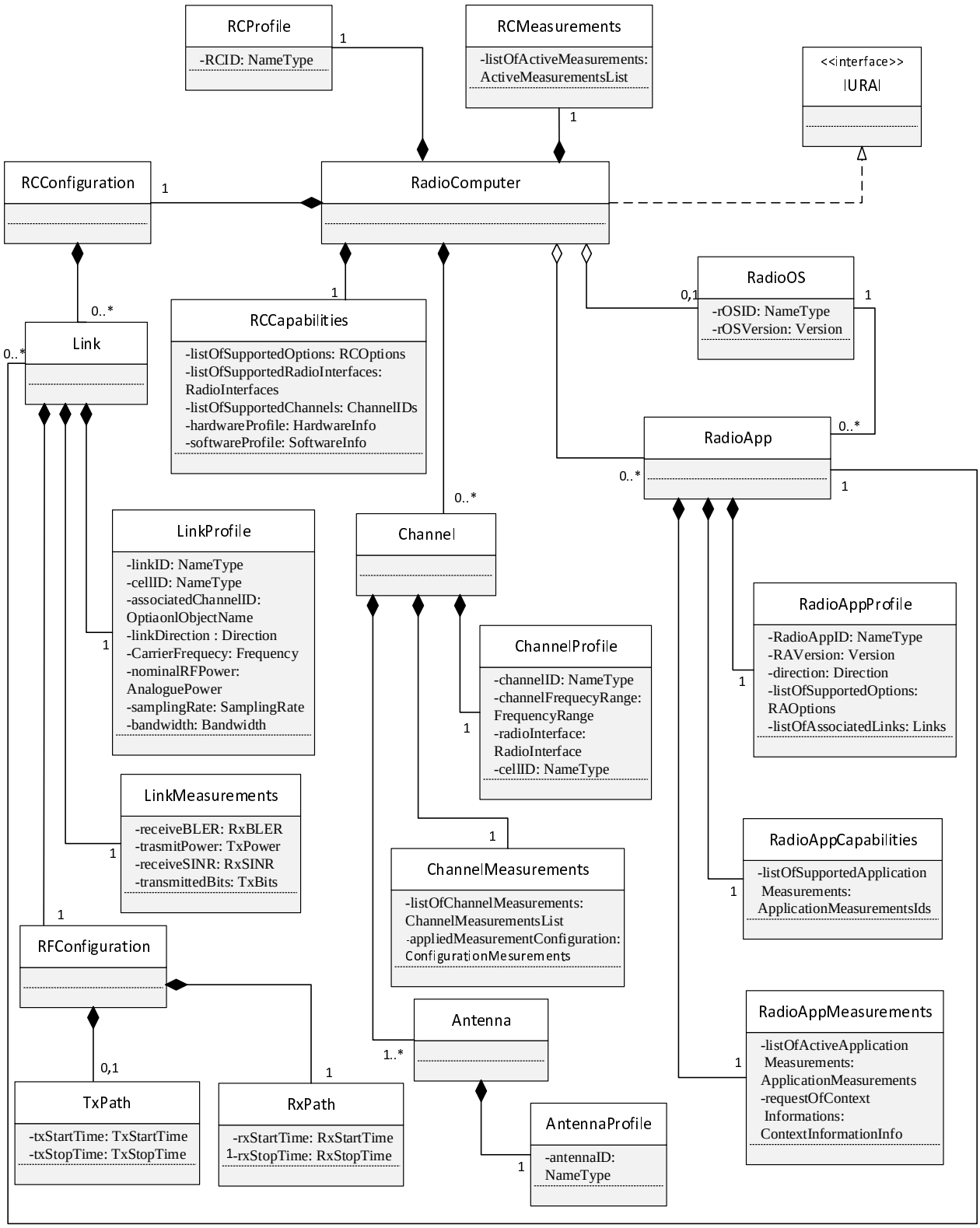

Figure 7.1: UML class diagram for Radio Computer classes related to URAI

The Radio Computer classes related to URAI are defined as follows:

- RadioComputer

This class contains all URA(s) related information about resources and interactions related to hardware and software of a reconfigurable MD, for example, computational/spectral resource usage, collection of context information, channel measurement results, etc.

- RadioOS

This class describes an installed Radio OS. Each instance of a RadioComputer class can relate to zero or one instance of RadioOS class (0,1). Each instance of RadioOS class is associated with zero or several instances of RadioApp class (0..*).

- RadioApp

This class describes one currently active Radio Application. Each instance of a RadioComputer class can relate to zero or several instances of RadioApp class (0..*). Each instance of RadioApp class is associated with one instance of Radio OS class. Each instance of RadioApp class is associated with zero, one or many instances of Link class

- RadioAppProfile

This class contains general information about the Radio Application, for example, Radio Application ID, current version of Radio Application, direction (downlink or uplink), links used to deliver this Radio Application, etc. Each instance of a RadioApp class can have only one instance of RadioAppProfile class as a member.

- RadioAppCapabilities

This class contains information about measurements supported by this Radio Application, for example, delay, loss, and bandwidth measurements. Each instance of a RadioApp class can have only one instance of RadioAppCapabilities class as a member.

- RadioAppMeasurements

This class contains measurements performed by this Radio Application, for example, delay, loss, and bandwidth measurements. Each instance of a RadioApp class can have only one instance of RadioAppMeasurements class as a member. Multiple measurements are contained within the instance of the class.

- RCProfile

This class contains general information about the Radio Computer, for example, terminal Identification (ID). Each instance of a "RadioComputer" class can have only one instance of RCProfile class as a member.

- RCMeasurements

This class contains current measurements (instantaneous measurement data and related metadata) related to Reconfigurable MD such as battery capacity, user mobility, MD location determination, and connection history information. Each instance of RadioComputer class shall have only one instance of RCMeasurements class as a member.

- RCCapabilities

This class contains information about Radio Computer capabilities including hardware, software, transmission and measurement capabilities such as supported RATs and maximum transmission power. Each instance of RadioComputer class shall have only one instance of RCCapabilities class as a member.

- Channel

This class contains one radio channel that may or may not be used by an active radio link. Each instance of RadioComputer class can have zero, one or several instances of Channel class as members (0..*). In case of an active radio link, at least one Channel class is available.

- ChannelProfile

This class contains general information about the radio channel such as channel ID, centre frequency, bandwidth, and used RAT. Each instance of Channel class shall have only one instance of Channel Profile class as a member.

- ChannelMeasurements

This class contains current measurements (instantaneous measurement data and related metadata) and the applied measurement configuration related to this radio channel such as interference and load measurements. Each instance of Channel class shall have only one instance of ChannelMeasurements class as a member.

- Antenna

This class contains information about antenna selection. Each instance of Channel class shall have at least one instance of Antenna class as a member. (1..*).

- AntennaProfile

This class contains general information about this antenna, such as antenna port, applicable frequency range and antenna gain. Each instance of Antenna class shall have only one instance of AntennaProfile class as a member.

- RCConfiguration

This class contains information about the current configuration of Radio Computer. Each instance of RadioComputer class shall have only one instance of RCConfiguration class as a member.

- Link

This class contains information about one active URA and the corresponding connection between the Reconfigurable MD and the Radio Access Network (RAN). Each instance of RCConfiguration class has zero, one or several instances of Link class as members (0..*). Each instance of Link class is associated with one instance of RadioApp class.

- LinkProfile

This class contains general information about this active connection, for example, link Identification (ID), serving cell ID, channel used, etc. Each instance of Link class shall have only one instance of LinkProfile class as a member.

- LinkMeasurements

This class contains current measurements (instantaneous measurement data and related metadata) related to this active connection, such as Block Error Rate (BLER), power, and Signal to Interference plus Noise Ratio (SINR) measurements. Each instance of Link class shall have only one instance of LinkMeasurements class as a member.

- RFConfiguration

This class contains information about the configuration of the RF transceiver. Each instance of Link class shall have only one instance of RFConfiguration class as a member.

- TxPath

This class contains information about one transmit path. Each instance of RFConfiguration class has zero or one instance of TxPath class as a member (0,1).

- RxPath

This class contains information about one receive path. Each instance of RFConfiguration class shall have only one instance of RxPath class as a member.

NOTE: The Channel Class is separate from the Link Class, but the Channel Measurements may be based on any MD configuration which may or may not be used for the final Link Configuration.

7.2 Class Definitions for Information Model

Each class of Radio Computer can be defined using the template presented in clause 6.1 and in accordance with the UML diagram of figure 7.1 which specifies the relations among all the classes of Radio Computer. Radio Computer classes defined in this way are shown in tables 7.1 to 7.21.

Table 7.1: RadioComputer Class

Class **RadioComputer**	
This class contains all URA related information about resources and interactions related to hardware and software of a reconfigurable MD.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	
CONTAINS	**RCCapabilities [1]**, **RCConfiguration [1], RCMeasurements [1], Channel [*], RCProfile [1], RadioAPP [*], RadioOS [0-1]**
SUPPORTED EVENTS	

Table 7.2: RadioOS Class

Class **RadioOS**			
This class describes installed Radio OS.			
DERIVED FROM			
ATTRIBUTES			
rOSID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of Radio OS.			
rOSVersion	*Value type:* Version	*Possible access:* Read	*Default value:* Not specified
This attribute describes a version of Radio OS.			
CONTAINED IN	**RadioComputer**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.3: RadioApp Class

Class **RadioApp**	
This class describes installed Radio Application.	
DERIVED FROM	
ATTRIBUTES	
This attribute contains a list of supported options.	
CONTAINED IN	**RadioComputer**
CONTAINS	**RadioAppProfile [1], RadioAppCapabilities [1], RadioAppMeasurements [1]**
SUPPORTED EVENTS	

Table 7.4: RadioAppProfile Class

Class **RadioAppProfile**			
This class contains general information about the Radio Application.			
DERIVED FROM			
ATTRIBUTES			
RadioAppID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of installed Radio Application.			
RAVersion	*Value type:* Version	*Possible access:* Read	*Default value:* Not specified
This attribute describes a version of Radio Application.			
direction	*Value type:* Direction	*Possible access:* Read	*Default value:* Not specified
This attribute describes whether this Radio Application is downlink or uplink application or both.			
listOfSupportedOptions	*Value type:* RAOptionsList	*Possible access:* Read	*Default value:* Not specified
This attribute contains a list of supported options (i.e. optional features as defined in related standard versus mandatory features).			
listOfAssociatedLinks	*Value type:* Links	*Possible access:* Read	*Default value:* Not specified
This attribute describes list of IDs of links used to transmit user data.			
CONTAINED IN	**RadioApp**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.5: RadioAppCapabilities Class

Class **RadioAppCapabilities**			
This class contains information about measurements supported by this Radio Application.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedApplication Capabilities	*Value type:* ApplicationMeasurementsIds	*Possible access:* Read	*Default value:* Not specified
This attribute describes capabilities supported by this Radio Application.			
CONTAINED IN	**RadioApp**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.6: RadioAppMeasurements Class

Class **RadioAppMeasurements**			
This class contains measurements performed by this Radio Application.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveApplication Measurements	*Value type:* ApplicationMeasurements	*Possible access:* Read-Add-Remove	*Default value:* Not specified
This attribute describes measurements that are currently performed by the Radio Application.			
requestOfContextInformations	*Value type:* *ContextInformationInfo*	*Possible access:* Read	*Default value:* Not specified
This attribute describes context informations that are requested by other entity such as Monitor. There can be a request for one-time delivery, a request for cyclic delivery or request for conditional delivery.			
CONTAINED IN	**RadioApp**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.7: RCProfile Class

Class **RCProfile**			
This class contains general information about the Radio Computer.			
DERIVED FROM			
ATTRIBUTES			
RCID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of radio computer.			
CONTAINED IN	**RadioComputer**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.8: RCMeasurements Class

Class **RCMeasurements**			
This class contains current measurements related to Reconfigurable Radio terminal.			
DERIVED FROM			
ATTRIBUTES			
listOfActiveMeasurements	*Value type:* ActiveMeasurementsList	*Possible access:* Read-Add-Remove	*Default value:* Not specified
This attribute describes a list of active measurements.			
CONTAINED IN	**RadioComputer**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.9: RCCapabilities Class

Class **RCCapabilities**			
This class contains information about Radio Computer capabilities including hardware, software, transmission and measurement capabilities.			
DERIVED FROM			
ATTRIBUTES			
listOfSupportedOptions	*Value type:* RCOptionsList	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a list of supported options.			
listOfSupportedRadioInterfaces	*Value type:* RadioInterfacesList	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes radio interfaces supported by this Radio Computer.			
listOfSupportedChannels	*Value type:* ChannelIDsList	*Possible access:* Read-Write	*Default value:* Not specified
This attributes describes frequency channels supported by this Radio Computer.			
hardwareProfile	*Value type:* HardwareInfo	*Possible access:* Read-Write	*Default value:* Not specified
This attributes describes hardware capabilities of this Radio Computer.			
softwareProfile	*Value type:* SoftwareInfo	*Possible access:* Read-Write	*Default value:* Not specified
This attributes describes software capabilities of this Radio Computer.			
CONTAINED IN	**RadioComputer**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.10: Channel Class

Class **Channel**	
This class describes one frequency channel that may or may not have active connections on it.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**RadioComputer**
CONTAINS	**ChannelProfile [1], ChannelMeasurements [1], Antenna [+]**
SUPPORTED EVENTS	

Table 7.11: ChannelProfile Class

Class **ChannelProfile**			
This class contains general information about this frequency channel.			
DERIVED FROM			
ATTRIBUTES			
channelID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of channel.			
channelFrequencyRange	*Value type:* FrequencyRange	*Possible access:* Read	*Default value:* Not specified
This attribute describes a value of channel frequency range.			
radioInterface	*Value type:* RadioInterface	*Possible access:* Read	*Default value:* Not specified
This attribute describes a radio interface.			
cellID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of connected cell.			
CONTAINED IN	**Channel**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.12: ChannelMeasurements Class

Class **ChannelMeasurements**			
This class contains current measurements related to this frequency channel.			
DERIVED FROM			
ATTRIBUTES			
listOfChannelMeasurements	*Value type:* ChannelMeasurementsList	*Possible access:* Read	*Default value:* Not specified
This attribute describes a list of channel measurements.			
appliedMeasurementsConfiguration	*Value type:* ConfigurationMeasurements	*Possible access:* Read	*Default value:* Not specified
This attribute describes configuration option of the MD, e.g. which Antenna(s) have been used, which RF front-end(s) have been used, etc.			
CONTAINED IN	**Channel**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.13: Antenna Class

Class **Antenna**	
This class contains information about antenna selection.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**Channel**
CONTAINS	**AntennaProfile [1]**
SUPPORTED EVENTS	

Table 7.14: AntennaProfile Class

Class **AntennaProfile**			
This class contains general information about this antenna.			
DERIVED FROM			
ATTRIBUTES			
antennaID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of antenna.			
CONTAINED IN	**Antenna**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.15: RCConfiguration Class

Class **RCConfiguration**	
This class contains information about the current configuration of Radio Computer.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**RadioComputer**
CONTAINS	**Link [*]**
SUPPORTED EVENTS	

Table 7.16: Link Class

Class **Link**	
This class contains information about one active Radio Application and corresponding connection between Reconfigurable Radio terminal and RANs.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**RCConfiguration**
CONTAINS	**LinkProfile [1], LinkMeasurements [1], RFConfiguration [1]**
SUPPORTED EVENTS	

Table 7.17: LinkProfile Class

Class **LinkProfile**			
This class contains general information about this active connection.			
DERIVED FROM			
ATTRIBUTES			
linkID	*Value type:* NameType	*Possible access:* Read	*Default value:* Not specified
This attribute describes ID of link about activated connection.			
cellID	*Value type:* NameType	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes ID connected cell.			
associatedChannelID	*Value type:* OptionalObjectName	*Possible access:* Read-Add-Remove	*Default value:* Not specified
This attribute describes ID of associated channel.			
linkDirection	*Value type:* Direction	*Possible access:* Read	*Default value:* Not specified
This attribute describes a direction of link.			
carrierFrequency	*Value type:* FrequencyRange	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a value of carrier frequency.			
nominalRFPower	*Value type:* AnaloguePower	*Possible access:* Read	*Default value:* Not specified
This attribute describes a value of nominal power.			
samplingRate	*Value type:* SamplingRate	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a value of sampling rate.			
Bandwidth	*Value type:* Bandwidth	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a value of bandwidth.			
CONTAINED IN	**Link**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.18: LinkMeasurements Class

Class **LinkMeasurements**			
This class contains current measurements related to this active connection.			
DERIVED FROM			
ATTRIBUTES			
receiveBLER	*Value type:* RxBLER	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a value of BLER for received data.			
transmitPower	*Value type:* TxPower	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a power of transmit signal.			
receiveSINR	*Value type:* RxSINR	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes a value of SINR for received data.			
transmittedBits	*Value type:* TxBits	*Possible access:* Read-Write	*Default value:* Not specified
This attribute describes transmitted bits.			
CONTAINED IN	**Link**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.19: RFConfiguration Class

Class **RFConfiguration**	
This class contains information about the configuration of RF transceiver.	
DERIVED FROM	
ATTRIBUTES	
CONTAINED IN	**Link**
CONTAINS	**TxPath [0-1], RxPath [1]**
SUPPORTED EVENTS	

Table 7.20: TxPath Class

Class **TxPath**			
This class describes one transmit path.			
DERIVED FROM	DERIVED FROM		
ATTRIBUTES			
txStartTime	*Value type:* TxStartTime	*Possible access:* Read-Write	*Default value:* Not specified
This attribute defines the time when the transceiver start transmission.			
txStopTime	*Value type:* TxStopTime	*Possible access:* Read-Write	*Default value:* Not specified
This attribute defines the time when the transceiver stop transmission.			
CONTAINED IN	**RFConfiguration**		
CONTAINS			
SUPPORTED EVENTS			

Table 7.21: RxPath Class

Class **RxPath**			
This class describes one receive path.			
DERIVED FROM	DERIVED FROM		
ATTRIBUTES			
rxStartTime	*Value type:* RxStartTime	*Possible access:* Read-Write	*Default value:* Not specified
This attribute defines the time when the transceiver start reception.			
rxStopTime	*Value type:* RxStopTime	*Possible access:* Read-Write	*Default value:* Not specified
This attribute defines the time when the transceiver stop reception.			
CONTAINED IN	**RFConfiguration**		
CONTAINS			
SUPPORTED EVENTS			

8 Interface Definition

8.1 Interface Overview

Figure 8.1 illustrates a UML diagram for URAI. URAI supports 3 basic services (i.e. Radio Application Management Services, User Data Flow Services, and Multiradio Control Services) which are further detailed in clauses 8.2, 8.3 and 8.4 respectively.

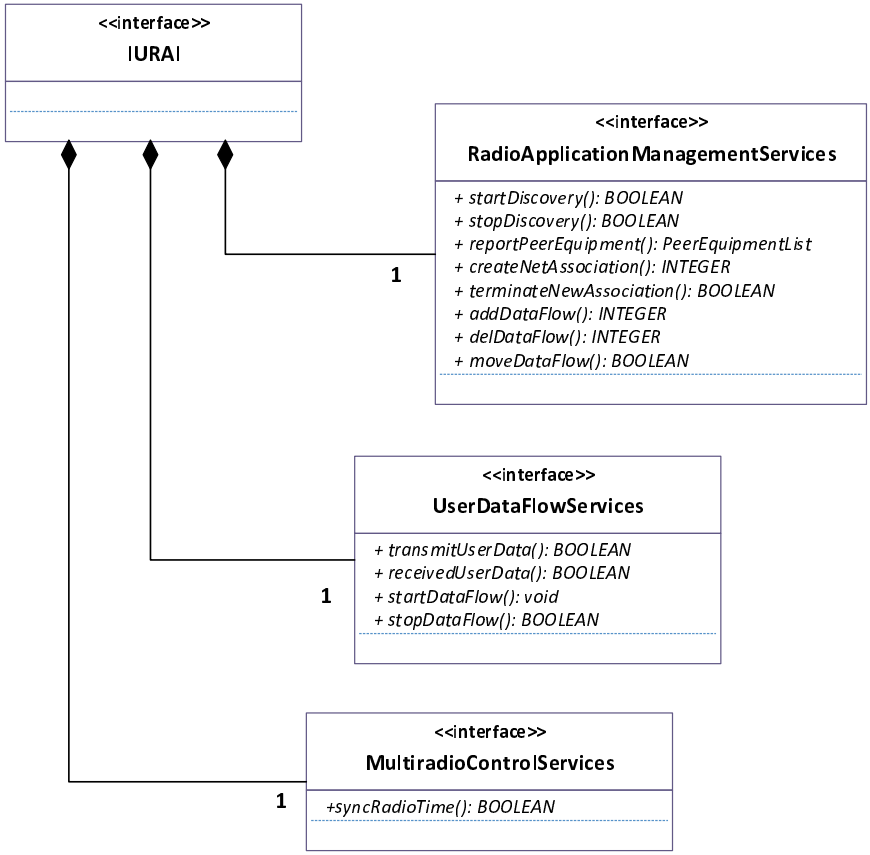

Figure 8.1: UML diagram for URAI

8.2 Radio Application Management Services

8.2.1 Overview on Radio Application Management Services

Table 8.1 describes an overview on Radio Application Management Services which are associated with RCM. Class definition and related operations are described in clause 8.5.

Table 8.1: Overview on Radio Application Management Services

Radio Application Management Services	Explanation
Report Discovered Peer Equipments.	URA reports RCM about the accessible peer equipment(s) found during the procedure of the discovery process. In the case when the reconfigurable MD is requested to be a peer equipment by another MD, the requesting MD shall be included as an accessible peer equipment too.
Create and Terminate Association with Peer Equipment.	RCM requests URA to create/terminate association with an accessible peer equipment.
Start and stop communication with Peer Equipment.	Among the activated and associated URAs, RCM requests some selected URAs to perform actual transmission/reception of user data.

8.2.2 Messages for Radio Application Management Services

The interfaces for Radio Application Management Services are used to transmit the following messages:

- From RCF to URA:
  - Request of report discovered Peer Equipments.
  - Request of create/terminate association with Peer Equipment.
  - Request of start and stop communication with Peer Equipment.
- From URA to RCF:
  - Confirmation of association creation.
  - Confirmation of association termination.
  - Confirmation of start communication with Peer Equipment.
  - Confirmation of stop communication with Peer Equipment.
  - Failure of association creation.
  - Failure of association termination.
  - Failure of start communication with Peer Equipment.
  - Failure of stop communication with Peer Equipment.
  - Information about discovered Peer Equipments.

8.3 User Data Flow Services

8.3.1 Overview on User Data Flow Services

Table 8.2 describes an overview on User Data Flow Services which are associated with FC. Class definition and related operations are described in clause 8.5.

Table 8.2: Overview on User Data Flow Services

Radio Application Management Services	Explanation
Send user data. The transmit user data are received by the FC and then transferred to the URA based on those information. Receive user dataURA indicates transfer of the received user data to FC.Data flow control.	In sending or receiving user data, there might be some conflicts in data flow between sender and receiver. URA requests FC to change the configuration of data flow.

8.3.2 Messages for User Data Flow Services

The interfaces for User Data Flow Services are used to transmit the following messages:

- From RCF to URA:
  - Request of user data transfer.
- From URA to RCF:
  - Request for changing of data flow configuration.
  - Information related to URA.
  - Confirmation of user data transfer.
  - Failure of user data transfer.
  - Information about user data.

8.4 Multiradio Control Services

8.4.1 Overview on Multiradio Control Services

Table 8.3 describes an overview on Multiradio Control Services which are associated with MRC. Class definition and related operations are described in clause 8.5.

Table 8.3: Overview on Multiradio Control Services

Multiradio Control Services	Explanation
Synchronize Radio Time	MRC request all the active URAs to operate with a unified synchronizm.

8.4.2 Messages for Multiradio Control Services

The interfaces for Multiradio Control Services are used to transmit the following messages:

- From RCF to URA:
  - Request of synchronize radio time.
- From URA to RCF:
  - Confirmation of synchronize radio time.
  - Failure of synchronize radio time.

8.5 Class Definitions for Interface

Each interface class related to URAI can be defined using the template presented in clause 6.2 and in accordance with the UML diagram of figure 8.1 which specifies the interface classes related to URAI. Tables 8.1 to 8.3 specify all the operations related to the three interface classes above described.

Table 8.4: RadioApplicationManagementServices Class

Class **RadioApplicationManagementServices**		
This class describes interfaces supporting Radio Application Management Services.		
OPERATIONS		
reprotPeerEquipment	*Return type:* PeerEquipmentList	*Value type:* public
This operation is needed for getting the list of discovered Peer Equipments.		
createNetAssociation	*Return type:* INTEGER	*Value type:* public
This operation is needed for creating an association with Peer Equipment.		
terminateNetAssociation	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for terminating an association.		
startTransmission	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for starting user data transmission with Peer Equipment.		
stopTransmission	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for stopping user data transmission with Peer Equipment.		
startReception	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for starting user data reception.		
stopReception	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for stopping user data reception.		

Table 8.5: UserDataFlowServices Class

Class **UserDataFlowServices**		
This class describes interfaces supporting User data Flow Services.		
OPERATIONS		
requestChangeofDataflow	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for requesting change of data flow.		
sendtoURA	*Return type:* UserData	*Value type:* public
This operation is needed for sending user data to URA.		
receivefromURA	*Return type:* UserData	*Value type:* public
This operation is needed for receiving user data from URA.		

Table 8.6: MultiradioControlServices Class

Class **MultiradioControlServices**		
This class describes interfaces supporting Multiradio Control Services.		
OPERATIONS		
syncRadioTime	*Return type:* BOOLEAN	*Value type:* public
This operation is needed for synchronizing radio time.		

Annex A (informative): Abstract Data Definitions

The following ASN.1 in Recommendation ITU-T X.680 [i.7] module contains all necessary abstract data definitions used in the attribute definitions in clause 7.2 and clause 8.5.

```
ETSI-TS-103-146-3-Type-Definitions DEFINITIONS ::= BEGIN
            ----------------------------------------------------------------
            ----------------------------------------------------------------
            -- START Common Data Types

            ----------------------------------------------------------------
            -- START Name Related Data Types

            NameType    ::= CHOICE      {
                    number      INTEGER,
                    string      PrintableString
            }

            ObjectName      ::= SEQUENCE OF NameType


            OptionalObjectName      ::= CHOICE  {
                    id          ObjectName,
                    void        NULL
            }

            ObjectNameList  ::= SEQUENCE OF ObjectName


            -- END Name Related Data Types
            ----------------------------------------------------------------

            ----------------------------------------------------------------
            -- START Version Related Data Types

            Version     ::= CHOICE  {
                intVersion          INTEGER,
                stringVersion       PrintableString
            }

            -- END Version Related Data Types
            ----------------------------------------------------------------

            -- END Common Data Types
            ----------------------------------------------------------------
            ----------------------------------------------------------------


            ----------------------------------------------------------------
            ----------------------------------------------------------------
            -- START Radio Application Related Data Types

            RAOptionID      ::= ENUMERATED      {
                lte5Mhz,lte10Mhz, lte20Mhz, ...
            }

            RAOptionsList       ::= SEQUENCE OF SEQUENCE        {
                rAOptionName        RAOptionID,
                rAOptionValue       ANY
            }

            RAMeasurementsID    ::= ENUMERATED      {
                observedDelay, observedDelayVariation, observedPacketLoss,
                observedBandwidth, ...
            }

            ApplicationMeasurements     ::= SEQUENCE OF SEQUENCE    {
                rAMeasurementsName      RAMeasurementsID,
                rAMeasurementsValue     ANY
            }

            ContextInformationID    ::= ENUMERATED  {
                BER, SNR, SINR, Output Power Levels, estimates of propagation delay, estimates of
                link attenuation, ...
```

```
}

ContextInformationList  ::= SEQUENCE OF SEQUENCE    {
    contextInformationName      ContextInformationID,
    contextInformationValue     ANY
}

SendingDuration     ::= ENUMERATED  {
    0,10ms,20ms, ...
}
SendingCondition        ::= CHOICE  {
    None, condition1, condition2, ...
}

ContextInformationInfo      ::= SEQUENCE OF SEQUENCE    {
    cIIInfo         ContextInformationList,
    period          SendingDuration
    condition       SendingCondition
}

--  END Radio Application Related Data Types
-----------------------------------------------------------------
-----------------------------------------------------------------


-----------------------------------------------------------------
-----------------------------------------------------------------
-- START Radio Computer Related Data Types

RadioApplicationIDList      ::= SEQUENCE OF OptionalObjectName


RCOptionID      ::= ENUMERATED      {
    mdrc-0, mdrc-1, mdrc-2, maximumTxPower, ...
}

RCOptionsList       ::= SEQUENCE OF SEQUENCE        {
    rCOptionName        RCOptionID,
    rCOptionValue       ANY
}

RadioInterfaceID        ::= ENUMERATED      {
    umts, hsdpa, wimax, lte, wifi, gsm, ...
}

RadioInterface      ::= CHOICE  {
    id          RadioInterfaceID,
    void        NULL
}

RadioInterfacesList     ::= SEQUENCE OF RadioInterfaceID

ChannelIDsList      ::= SEQUENCE OF OptionalObjectName

HardwareInfo        ::= ENUMERATED      {
    fixedPipeline, programmablePipeline, hybridPipeline, ...
}

SoftwareInfo        ::= ENUMERATED      {
    rOSVersion, compiler, ...
}

Direction       ::= ENUMERATED      {
    downlink, uplink
}

RxBLER      ::= SEQUENCE    {
    accBLER     REAL,
    period      REAL OPTIONAL,
    instBLER    REAL OPTIONAL
}

TxPower     ::=     SEQUENCE    {
    power       REAL,
    unit        CHARACTER
}

RxSINR      ::= SEQUENCE    {
```

```
    accSINR     REAL,
    period      REAL OPTIONAL,
    instSINR    REAL OPTIONAL
}

Links       ::= SEQUENCE OF OptionalObjectName

TxBits      ::= SEQUENCE    {
    transmittedBit      REAL,
    unit                CHARACTER
}


ActiveMeasurementID ::= ENUMERATED      {
    transmitPower, transportLoad, processingLoad, ...
}

ActiveMeasurementIDs        ::= SEQUENCE OF {
    activeMeasurementID
}

ActiveMeasurementsList      ::=     SEQUENCE OF SEQUENCE    {
    activeMeasurementName       ActiveMeasurementID,
    activeMeasurementValue      ANY
}


FrequencyRange      ::= SEQUENCE    {
    centralFrequency        REAL,
    frequencyBand           REAL
}

AnaloguePower       ::=     SEQUENCE    {
    power       REAL,
    unit        CHARACTER
}

SamplingRate        ::=     SEQUENCE    {
    samplingRate        REAL,
    unit        CHARACTER
}

Bandwidth       ::=     SEQUENCE    {
    bandWidth       REAL,
    unit        CHARACTERs
}

TxStartTime         ::=         CHOICE  {
    absoluteTime    GeneralizedTime,
    relativeTime    INTEGER
}

TxStopTime          ::=         CHOICE  {
    Undefined       NULL,
    absoluteTime    GeneralizedTime,
    relativeTime    INTEGER
}


RxStartTime         ::=         CHOICE  {
    absoluteTime    GeneralizedTime,
    relativeTime    INTEGER
}


RxStopTime          ::=         CHOICE  {
    Undefined       NULL,
    absoluteTime    GeneralizedTime,
    relativeTime    INTEGER
}


ChannelMeasurementID        ::= ENUMERATED  {
    channelInterference, channelLoad, ...
}

ChannelMeasurementsList     ::= SEQUENCE OF SEQUENCE    {
    channelMeasurementName      ChannelMeasurementID,
```

```
    channelMeasurementValue     ANY
}

ConfigurationMeasurements::=    ENUMERATED  {
    antennaProt, RFfrontend, ...
}


-- END Radio Computer Related Data Types
-----------------------------------------------------------------
-----------------------------------------------------------------


-----------------------------------------------------------------
-----------------------------------------------------------------
-- START Unified Radio Application Interface Related Data Types

RadioAppParameterID ::= ENUMERATED {
    A, b, c, ...
}

RadioAppParameters      ::= SEQUENCE OF SEQUENCE    {
    radioAppParameterName   RadioAppParameterID,
    radioAppParameterValue  ANY
}


RadioAppsList       ::= SEQUENCE OF SEQUENCE    {
    RadioAppID          INTEGER,
    RadioAppName        PrintableString
}


RadioMeasurementID      ::= ENUMERATED {
    A, B, C, ...
}

RadioMeasurementsList   ::= SEQUENCE OF SEQUENCE {
    radioMeasurementName        RadioMeasurementID,
    radioMeasurementValue       ANY
}

PeerEquimentId          ::= SEQUENCE OF OptionalObjectName

PeerEquipmentList       ::= SEQUENCE OF {
    PeerEquipmentId
}

UserData        ::= SEQUENCE OF {
    userDataID          INTEGER,
    userDataValue       OBJECT
}


-- END Unified Radio Application Interface Related Data Types
-----------------------------------------------------------------
-----------------------------------------------------------------
```

Annex B (informative): URAI Qualification Methods for Validation

The URAI requirements are basis for qualification methods to validate that the requirements can be met. A feature list exposing URAI capabilities is created. Qualification methods correspond to the feature list and they qualify features of a particular URAI implementation against the feature list.

The following qualification methods might be typically applied:

- Demonstration - The operation of interfacing entities that rely on observable functional operation.
- Test - The operation of interfacing entities using specialist test equipment to collect data for analysis.
- Analysis - The processing of data obtained from methods, such as reduction, interpretation, or extrapolation of test results.
- Inspection - The visual examination of interfacing entities, documentation, etc.
- Special qualification methods - Methods for the interfacing entities, such as specialist tools, techniques, procedures, facilities, etc.

History

Document history		
V1.1.1	February 2016	Publication